EE5103 CA 2

Guo Shiping
A0260014Y

Q.1

a)

$$H(z) = \frac{z+0.9}{z^2-2.5z+1} = \frac{B(z)}{A(z)}$$

The zero is stable, we want wo cancel it, let

$$\begin{aligned} R(z) &= z+0.9 \\ S(z) &= s_0 z + s_1 \\ A_o(z) &= B(z) = z+0.9 \end{aligned}$$

$$\begin{aligned} A(z)R(z) + B(z)S(z) &= A_m(z)A_o(z) \\ (z^2-2.5z+1)(z+0.9) + (z+0.9)(s_0 z + s_1) &= (z^2-1.8z+0.9)(z+0.9) \\ z^2 + (s_0-2.5)z + s_1 + 1 &= z^2 - 1.8z + 0.9 \end{aligned}$$

$$\Rightarrow \begin{cases} s_0 - 2.5 = -1.8 \\ s_1 + 1 = 0.9 \end{cases}$$

$$\Rightarrow \begin{cases} s_0 = 0.7 \\ s_1 = -0.1 \end{cases}$$

So, $(z) = 0.7z - 0.1$

$$\frac{Y(z)}{U_c(z)} = \frac{T(z)B(z)}{A(z)R(z)+B(z)S(z)} = \frac{T(z)B(z)}{A_m(z)A_o(z)} = \frac{T(z)}{A_m(z)}$$

Besides, the steady-state gain should be one, and the zero has been canceled, so

$$\begin{aligned} \frac{T(1)}{A_m(1)} &= 1 \\ T(z) &= A_m(1) = 0.1 \\ \frac{Y(z)}{U_c(z)} &= \frac{0.1}{z^2-1.8z+0.9} \end{aligned}$$

The controller can be expressd as

$$(q+0.9)u(k) = 0.1u_c(k) - (0.7q-0.1)y(k)$$

b)

Cause we don't need cancel zero, so let

$$\begin{aligned} R(z) &= z + r_1 \\ S(z) &= s_0 z + s_1 \\ A_o(z) &= z \end{aligned}$$

$$\begin{aligned} A(z)R(z) + B(z)S(z) &= A_m(z)A_o(z) \\ (z^2-2.5z+1)(z+r_1) + (z+0.9)(s_0 z + s_1) &= z(z^2-1.8z+0.9) \\ z^3 + (r_1 - 2.5 + s_0)z^2 + (1 - 2.5r_1 + 0.9s_0 + s_1) + (r_1 + 0.9s_1) &= z^3 - 1.8z^2 + 0.9z \end{aligned}$$

$$\Rightarrow \begin{cases} r_1 - 2.5 + s_0 = -1.8 \\ 1 - 2.5r_1 + 0.9s_0 + s_1 = 0.9 \\ r_1 + 0.9s_1 = 0 \end{cases}$$

$$\Rightarrow \begin{cases} r_1 = 0.162 \\ s_0 = 0.538 \\ s_1 = -0.18 \end{cases}$$

So, $R(z) = z - 0.162,\ \ S(z) = 0.538z - 0.18$ The steady-state gain should be one, and we want to decrease the order of process

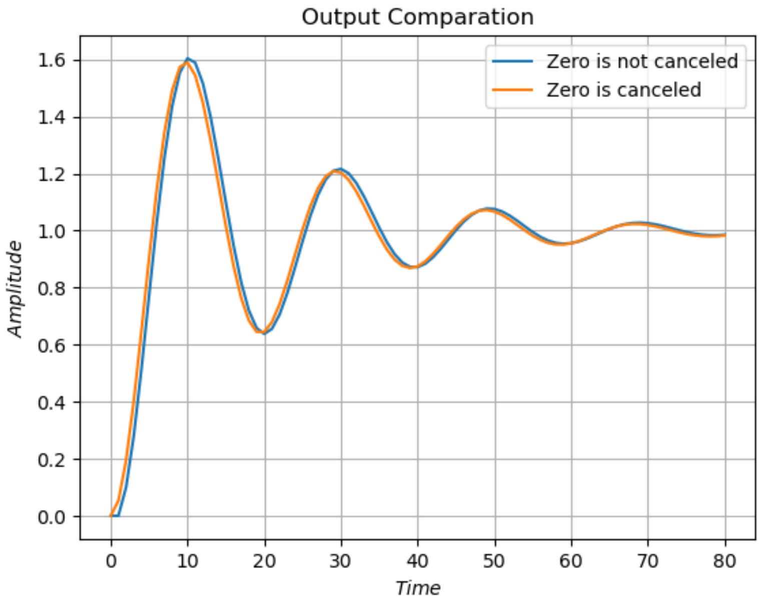
$$\begin{cases} T(z) = t_o A_o(z) \\ \frac{T(1)B(1)}{A_m(1)A_o(1)} = 1 \end{cases}$$

$$\Rightarrow T(z) = \frac{1}{19} z = 0.0526z$$

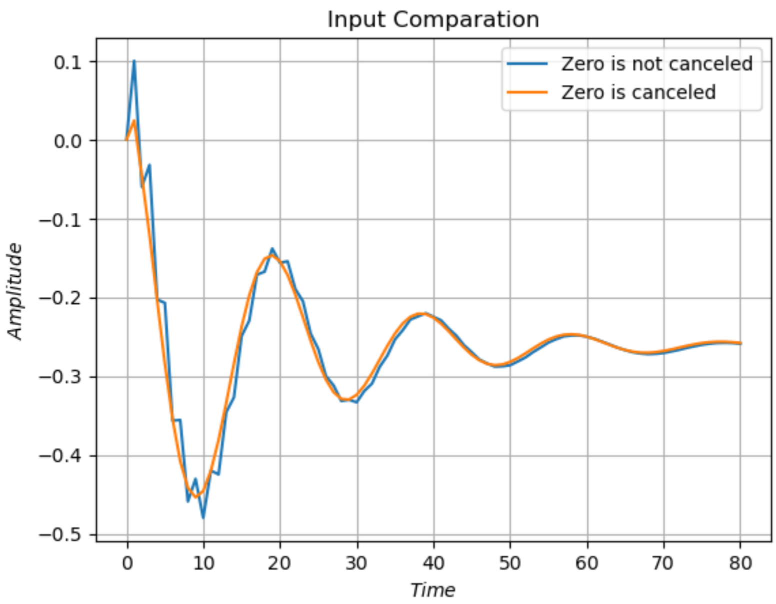
$$\frac{Y(z)}{U_c(z)} = \frac{T(z)B(z)}{A_m(z)A_o(z)} = \frac{0.0526z + 0.047}{z^2 - 1.8z + 0.9}$$

In [17]:
```
import control as ct
import numpy as np
import matplotlib.pyplot as plt
```

In [52]:
```
H1 = ct.tf([0.1],[1, -1.8, 0.9], 1)
H2 = ct.tf([0.0526, 0.047],[1, -1.8, 0.9], 1)

t1, y1 = ct.step_response(H1, 80)
t2, y2 = ct.step_response(H2, 80)
plt.plot(t1,y1)
plt.plot(t2,y2)
plt.title('Output Comparation')
plt.xlabel('$Time$')
plt.ylabel('$Amplitude$')
plt.legend(['Zero is not canceled','Zero is canceled'])
plt.grid(True)
```



In [58]:
```
t = t1
N = np.size(t1)

ua = np.zeros([1,N])
ub = np.zeros([1,N])

for i in range(0, N-1):
    ua[0, i+1] = 0.1 - 0.7*y1[i+1] + 0.1*y1[i] - 0.9*ua[0,i]
    ub[0, i+1] = 0.0526 - 0.538*y2[i+1] + 0.18*y2[i] - 0.162*ub[0,i]
plt.figure(1)
plt.plot(t, ua[0, :])
plt.plot(t, ub[0, :])
plt.title('Input Comparation')
plt.xlabel('$Time$')
plt.ylabel('$Amplitude$')
plt.legend(['Zero is not canceled','Zero is canceled'])
plt.grid(True)
```



The results demonstrate that the "zero cancellation" control approach exhibits slightly superior performance compared to the "zero not canceled" method. This superiority is attributed to the former method's faster response and the smoother, more manageable nature of the input signal.

Q.2

a)

Let

$$A(z) = z^2 - 4z, +4$$
$$B(z) = z - 0.8$$
$$A_m(z) = z^2$$
$$B_m(z) = 1$$

Because $B(z)$ is stable, and we want the close-loop transfer function close to the reference model, and reject constant disturbance.

So, let

$$R(z) = (z-1)B(z)$$
$$S(z) = s_0 z^2 + s_1 z + s_2$$
$$A_o(z) = zB(z)$$

we can get:

$$A(z)R(z) + B(z)S(z) = A_{cl}(z) = A_o(z)A_m(z)$$
$$(z^2 - 4z + 4)(z-1)B(z) + (s_0 z^2 + s_1 z + s_2)B(z) = z^3 B(z)$$
$$z^3 + (s_0 - 5)z^2 + (s_1 + 8)z + (s_2 - 41) = z^3$$

$$\Rightarrow \begin{cases} s_0 - 5 = 0 \\ s_1 + 8 = 0 \\ s_2 - 4 = 0 \end{cases}$$
$$\Rightarrow \begin{cases} s_0 = 5 \\ s_1 = -8 \\ s_2 = 4 \end{cases}$$

We can get:

$$R(z) = (z-1)(z-0.5)$$
$$S(z) = 5z^2 - 8z + 4$$

The close-loop transfer function is:

$$G(z) = \frac{T(z)B(z)}{A(z)R(z) + B(z)S(z)} = \frac{T(z)}{z^3}$$

Because we want the close-loop transfer function as close to the reference model as possible, so let $T(z) = z$, the controller can be expressed as:

$$(q-1)(q-0.5)u(k) = qu_c(k) - (5q^2 - 8q + 4)y(k)$$

b)

We want to reject constant disturbance, so let

$$R(z) = (z-1)(z+r_1)$$
$$S(z) = s_0 z^2 + s_1 z + s_2$$
$$A_o(z) = z^2$$

we can get:

$$A(z)R(z) + B(z)S(z) = A_{cl}(z) = A_o(z)A_m(z)$$
$$(z^2 - 4z + 4)(z-1)(z+r_1) + (s_0 z^2 + s_1 z + s_2)(z-0.5) = z^4$$
$$z^4 + (r_1 + s_0 - 5)z^3 + (-5r_1 - 0.5s_0 + s_1 + 8)z^2 + (4r_1 + 0.5s_2)z + (r_1 - 5 + s_0) = z^4$$
$$\Rightarrow \begin{cases} 8 - 5r_1 + s_1 - 0.5s_0 = 0 \\ 8r - 4 + s_2 - 0.5s_1 = 0 \\ 4r_1 + 0.5s_2 = 0 \\ r_1 - 5 + s_0 = 0 \end{cases}$$
$$\Rightarrow \begin{cases} r_1 = -\frac{5}{9} \\ s_0 = 4.44 \\ s_1 = -3 \\ s_2 = -4.44 \end{cases}$$

So,

$$R(z) = (z-1)(z-\frac{5}{9}),\ S(z) = 4.44z^2 - 3z - 4.44$$
$$U_{fb}(z) = -\frac{S(z)}{R(z)}Y(z) = -\frac{4.44z^2 - 3z - 4.44}{(z-1)(z-\frac{5}{9})}Y(z)$$

the transfer function changes to:

$$G(z) = H_{ff}(z)\frac{B(z)R(z)}{A(z)R(z) + B(z)S(z)} = H_{ff}\frac{B(z)R(z)}{A_{cl}(z)}$$

We want the close-loop transfer function as close to the reference model as possible, so:

$$H_{ff}\frac{B(z)R(z)}{A_{cl}(z)} = \frac{B_m(z)}{A_m(z)}$$
$$H_{ff} = \frac{A_o(z)B_m(z)}{B(z)R(z)} = \frac{z^2}{(z-1)(z-\frac{5}{9})(z-0.5)}$$
$$U(z) = -\frac{4.44z^2 - 3z - 4.44}{(z-1)(z-\frac{5}{9})}Y(z) + \frac{z^2}{(z-1)(z-\frac{5}{9})(z-0.5)}U_c(z)$$

Q.3

Dynamics of the vehicle:

$$m\ddot{y} + b\dot{y} = u(t)$$
$$1000\ddot{y} + 200\dot{y} = u(t)$$

$$T(s) = \frac{0.001}{s^2 + 0.2s}$$

1. The overshoot is less than 10%.
2. The settling time is less than 10 s.
3. The controller can reject the influence of unknown constant distur- bance. To satisfy these requirements.

$$e^{-\frac{\pi\zeta}{\sqrt{1-\zeta^2}}} \leq 10$$
$$\zeta = 0.591$$
$$w_n = \frac{4.6}{t_s\zeta},\ t_s \leq 10s$$
$$w_n = 0.7783$$

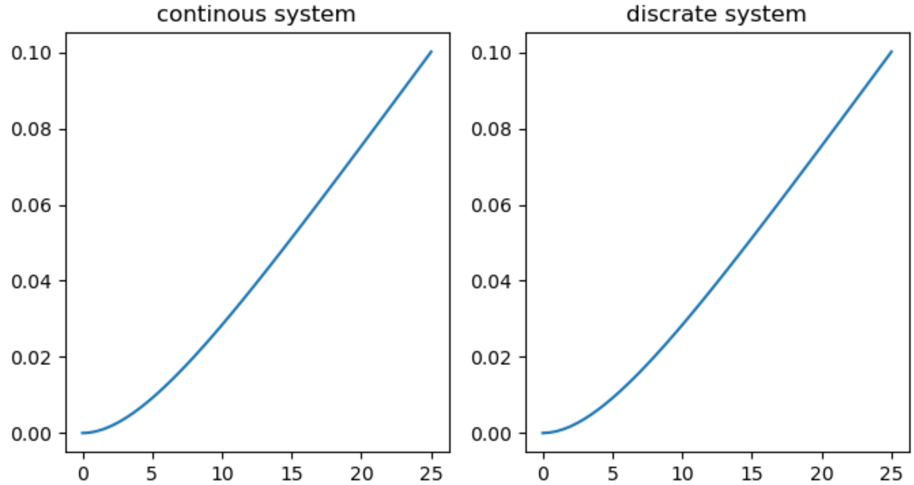
reference model is:

$$H(s) = \frac{w_n^2}{s^2 + 2\zeta w_n s + w_n^2} = \frac{0.6058}{s^2 + 0.92s + 0.6058}$$

After sampling the discrate time transfer function are:

In [66]:
```
# System transfer function
Hc = ct.tf([0.001],[1,0.2,0])
tc, yc = ct.step_response(Hc)
plt.figure(1, figsize = (8,4))
plt.subplot(1,2,1)
plt.plot(tc, yc)
plt.title("continous system")

#Hc
Hd = ct.c2d(Hc, 0.5)
td, yd = ct.step_response(Hd)
plt.subplot(1,2,2)
plt.plot(td, yd)
plt.title("discrate system")
```

Out[66]: `Text(0.5, 1.0, 'discrate system')`



In [55]:
```
print("The discrate system transfer function is:")
Hd
```

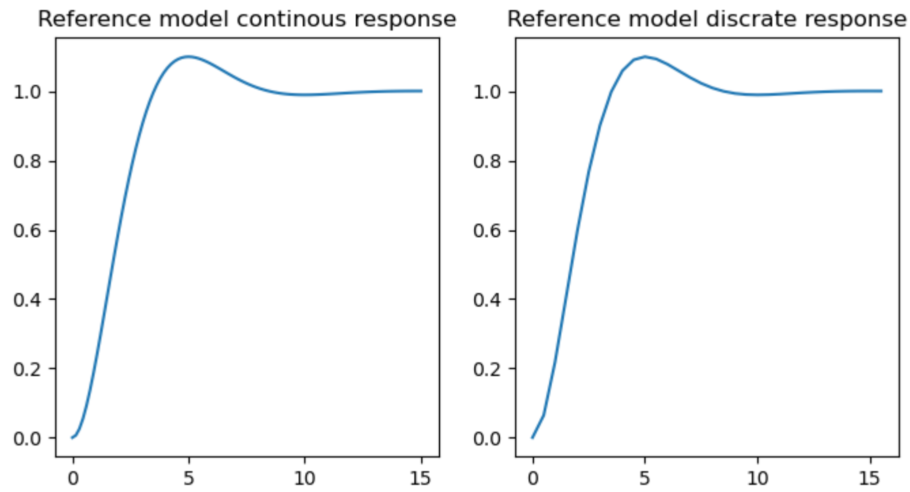
```
The discrate system transfer function is:
```

Out[55]:

$$\frac{0.0001209z + 0.000117}{z^2 - 1.905z + 0.9048} \quad dt = 0.5$$

In [65]:
```
# reference model
wn = 0.7783
thita = 0.591
Hrs = ct.tf([wn**2],[1, 2*thita*wn, wn**2])
trc, yrc = ct.step_response(Hrs)
plt.figure(1, figsize = (8,4))
plt.subplot(1,2,1)
plt.plot(trc, yrc)
plt.title("Reference model continous response")

Hrd = ct.c2d(Hrs, 0.5)
trd, yrd = ct.step_response(Hrd)
plt.subplot(1,2,2)
plt.plot(trd, yrd)
plt.title("Reference model discrate response")
```

Out[65]:
```
Text(0.5, 1.0, 'Reference model discrate response')
```



In [57]:
```
print("The discrate reference model transfer function is:")
Hrd
```

```
The discrate reference model transfer function is:
```

Out[57]:

$$\frac{0.06454z + 0.05533}{z^2 - 1.511z + 0.6313} \quad dt = 0.5$$

Let

$$A(z) = z^2 - 1.905z + 0.9048$$
$$B(z) = 0.0001209z + 0.000117$$
$$A_m(z) = z^2 - 1.511z + 0.6313$$
$$B_m(z) = 0.06454z + 0.05533$$

Because $B(z)$ is stable, and we want the close-loop transfer function close to the reference model, and reject constant disturbance. So, let

$$R(z) = (z-1)B(z)$$
$$S(z) = s_0 z^2 + s_1 z + s_2$$
$$A_o(z) = zB(z)$$

we can get:

$$A(z)R(z) + B(z)S(z) = A_{cl}(z) = A_o(z)A_m(z)$$
$$(z^2 - 1.905z + 0.9048)(z-1)B(z) + (s_0 z^2 + s_1 z + s_2)B(z) = zB(z)(z^2 - 1.511z + 0.6313)$$
$$\Rightarrow \begin{cases} s_0 - 2.905 = -1.511 \\ s_1 + 0.9048 + 1.905 = 0.6313 \\ s_2 - 0.9048 = 0 \end{cases} \Rightarrow \begin{cases} s_0 = 1.394 \\ s_1 = -2.1785 \\ s_2 = 0.9048 \end{cases}$$

The close-loop transfer function is:

$$G(z) = \frac{T(z)B(z)}{A(z)R(z) + B(z)S(z)} = \frac{B_m}{A_m}$$

Because we want the close-loop transfer function as close to the reference model as possible, so let $T(z) = 0.06454z + 0.05533$, the controller can be expressed as:

$$R(q)u(k) = T(q)u_c(k) - S(q)y(k)$$
$$(q-1)(0.0001209q + 0.000117)u(k) = (0.06454q + 0.05533)u_c(k) - (1.394q^2 - 2.1785q + 0.9048)y(k)$$

Q.4

a)

We want to use $u(k)$ to control the output signal. So, we need to convert the equation so that it contains the input $u(k)$:

$$y(k+1) = y(k) + \frac{cu(k-1)}{y^2(k-1)+1}$$

$$y(k+2) = y(k+1) + \frac{cu(k)}{y^2(k)+1}$$

$$y(k+2) = y(k) + \frac{cu(k-1)}{y^2(k-1)+1} + \frac{cu(k)}{y^2(k)+1}$$

Then let r(k+2)=y(k+2), the input signal can be expressed as:

$$u(k) = \frac{1}{c}[r(k+2) - y(k+1)][y^2(k)+1]$$

$$= \frac{1}{c}[r(k+2) - y(k) - \frac{cu(k-1)}{y^2(k-1)+1}]$$

b)

I have no idea how to do the linealization for this equation.

I test it by coding and found the output is bounded and perfect tracking seems not be affected by c too much as long as c is not 0.

In [80]:

```
import numpy as np
c = [-1, -0.5, 0.5, 1, 5, 100]
plt.figure(1, figsize = (10,7))
for k in range(0, 6):
    t = np.linspace(1,100000,1000000)
    y = np.zeros([1, 1000000])
    u = np.ones([1,1000000])
    for i in range(1,999997):
        y[0, i+2] = y[0, i+1] + c[k]*u[0, i-1]/(y[0,i-1]**2 + 1) + c[k]*u[0, i]/(y[0, i]**2+1)
    plt.subplot(2,3,k+1)
    plt.title('c = %f'% c[k])
    plt.plot(t[10:-2], y[0,10:-2])
```